Claims

1. - 10. (cancelled)

11. (Currently amended) A method for controlling functional units in a processor, the method comprising:

a configuration step, in which a sequence of primary instruction words consisting of multiple instruction word parts and originating from a translation of a program code is compressed and stored as a sequence of related program ~~words,~~ words; and

~~according to which, in~~ a subsequent execution step,

wherein sequential secondary instruction words consisting of a plurality of instruction word parts and having the full instruction word width needed to control all functional units are generated from the sequence of related program words,

wherein the configuration step is configured so that a program word (7) has a first characteristic (10) of a primary instruction word (5) from a first reference group of preceding primary instruction ~~words,~~ words (5), which has the greatest similarity to the primary instruction word (5) associated with the program ~~word,~~ word (7), and contains instruction word parts (6) that differentiate the primary instruction word belonging to the program word from the primary instruction word belonging to the first ~~characteristic;~~ characteristic (10), and

wherein in the subsequent execution step ~~further:~~ comprises:

~~obtaining a store of~~ storing a second group (12) of secondary instruction words (9) corresponding in number to the first group (11),

wherein each secondary instruction word is provided with a second ~~characteristic;~~ characteristic (13),

wherein using ~~in accordance with~~ the first characteristic (10) contained in the program ~~word,~~ word (7) ~~ascertaining that~~ a particular secondary instruction word (9) from the second group (12) is ~~corresponds to the~~ associated with the primary instruction word (5) via the associated second ~~characteristic;~~ characteristic (13), and

~~generating a specific~~ wherein the particular secondary instruction word (9) ~~controlling functional units~~ corresponding to the program word (7) is generated ~~such~~ so that the instruction word parts (6) contained in the program word (7) are exchanged in ~~combined with~~ the particular secondary instruction word (9) from the second ~~group.~~ group (12); and

obtaining a store of the second group of secondary instruction words,

wherein the configuration step consists forming and storing previously generated complete Very Long Instruction Words as the first reference group of preceding primary instruction words and the subsequent execution step comprises dynamic updating of the stored reference group of complete Very Long Instruction Words.

12. (Currently amended) The method in accordance with claim 11, wherein:

the first group (11) consists of a first number of primary instruction words (5) that directly precede the primary instruction word (5) associated with ~~belonging to~~ the program word (7); ~~word;~~ and

the second group (12) consists of a second number of secondary instruction words (9) that is at least equal to the first number, where, prior to the generation of the next sequential secondary instruction word (9), ~~word,~~ each most recent secondary instruction word (9), ~~word,~~ is appended to the second group (12) as the last word, and the first secondary instruction word (9) to have been added and that is in excess of the second number is removed from the second group (12). ~~group.~~

13. (Currently amended) The method in accordance with claim 11, further comprising replacing the particular secondary instruction word (9) in the second group (12) with the newly generated specific secondary instruction word (9) in that the latter is stored in place of the secondary instruction word that was used for its generation. ~~word.~~

14. (Currently amended) The method in accordance with claim 12, [[~~11,~~]] further comprising excluding the newly generated specific secondary instruction word (9) from the store of the second group of secondary instruction words (12). ~~words.~~

15. (Currently amended) The method in accordance with claim 11, wherein the first characteristic (10) is formed as a minimum code distance between the primary instruction word belonging to the current program word ~~in question~~ and the primary instruction word with the greatest similarity.

16. (Currently amended) The method in accordance with 11, wherein the second characteristic (13) comprises an address corresponding to the first characteristic that is the address of a preceding secondary instruction word in a memory (14) that is used for storage of the second group (12). ~~group.~~

17. (Currently amended) The method in accordance with claim 11, wherein the program word (7) comprises a number of instruction word parts (6) ~~to be differentiated, that instruction word part~~ that occur ~~occurs most~~ frequently in the configuration step, and ~~and~~ wherein a plurality of program words (7) are used to assemble secondary instruction words (9) that require more than the number of instruction words (9) stored in one program word (7) for the secondary instruction word (9) used for generation.

18. (Currently amended) The method in accordance with claim 17, [[~~11,~~]] wherein the instruction word parts (6) are compressed in one program word (7) by reducing the bit width to the extent that it is possible to represent the most frequently occurring instruction word parts (6), ~~parts,~~ and wherein multiple program words are used when instruction word parts (6) occur that require a greater bit width in order to be represented.

19. (Currently amended) The method in accordance with claim 18, wherein the width of the instruction word parts (6) in the program word (7) is reduced by half,

~~halved,~~ and up to two program words are provided for representation of the instruction word parts (6). ~~parts.~~

20. (Currently amended) A processor arrangement for carrying out the method of claim 11, comprising:

a plurality of functional units;

an instruction word memory associated with the functional units; and

an instruction word buffer for storing instruction words that have already been generated and have a width that is at least the size of the bit width of the secondary instruction word, ~~word;~~ the instruction word buffer including a memory (14) with selective line-by-line access.